Fix an alphabet $\Sigma$. For $x, w \in \Sigma^*$ we let $x \leq w$ denote the condition that $x$ is a subsequence of $w$. For a language $L \subseteq \Sigma^*$, define

$$\text{SUBSEQ}(L) := \{ x \in \Sigma^* \mid (\exists w \in L) x \leq w \}.$$

**Theorem 1.** $\text{SUBSEQ}(L)$ is regular for any $L \subseteq \Sigma^*$.

Clearly, $\text{SUBSEQ}(\text{SUBSEQ}(L)) = \text{SUBSEQ}(L)$ for any $L$, since $\leq$ is transitive. We’ll say that $L$ is $\leq$-closed if $L = \text{SUBSEQ}(L)$. So Theorem 1 is equivalent to the statement that a language $L$ is regular if $L$ is $\leq$-closed. The remainder of this note is to prove Theorem 1.

## 1 Preliminaries

We let $\mathbb{N} = \omega = \{ 0, 1, 2, \ldots \}$ be the set of natural numbers. We will assume WLOG that all symbols are elements of $\mathbb{N}$ and that all alphabets are finite, nonempty subsets of $\mathbb{N}$. We can also assume WLOG that all languages are nonempty. We extend the star notation to $\mathbb{N}$, letting $\mathbb{N}^*$ be the set of all finite strings over $\mathbb{N}$.

For a finite set $X$ we let $|X|$ denote the cardinality of $X$.

**Definition 2.** For any alphabet $\Sigma = \{ n_1 < \cdots < n_k \}$, we define the canonical string for $\Sigma$,

$$\sigma_\Sigma := n_1 \cdots n_k,$$

the concatenation of all symbols of $\Sigma$ in increasing order. If $w \in \Sigma^*$, we define the number

$$\ell_\Sigma(w) := \max\{ n \in \mathbb{N} \mid (\sigma_\Sigma)^n \leq w \}.$$

**Observation 3.** $(\sigma_\Sigma)^n$ has any string in $\Sigma^*$ of length at most $n$ as a subsequence. Thus for any string $w$ and $x \in \Sigma^*$, if $|x| \leq \ell_\Sigma(w)$, then $x \leq w$.

Our regular expressions (regexps) are built from the atomic regexps $\varepsilon$ and $a \in \mathbb{N}$ using union, concatenation, and Kleene closure in the standard way (we omit $\emptyset$ as a regexp since all our languages are nonempty). For regexp $r$, we let $L(r)$ denote the language of $r$. We consider regexps as syntactic objects, distinct from their corresponding languages. So for regexps $r$ and $s$, by saying that $r = s$ we mean that $r$ and $s$ are syntactically identical, not just that $L(r) = L(s)$. For any alphabet $\Sigma = \{ n_1, \ldots, n_k \} \subseteq \mathbb{N}$, we let $\Sigma$ also denote the regexp $n_1 \cup \cdots \cup n_k$ as usual, and in keeping with our view of regexps as syntactic objects, we will heretofore be more precise and say, e.g., “$L \subseteq L(\Sigma^*)$” rather than “$L \subseteq \Sigma^*$.”

**Definition 4.** A regexp $r$ is primitive syntactically $\leq$-closed (PSC) if $r$ is one of the following two types:

**Bounded:** $r = a \cup \varepsilon$ for some $a \in \mathbb{N}$;

**Unbounded:** $r = \Sigma^*$ for some alphabet $\Sigma$. 

1
The rank of such an \( r \) is defined as
\[
\text{rank}(r) := \begin{cases} 
0 & \text{if } r \text{ is bounded,} \\
|\Sigma| & \text{if } r = \Sigma^*.
\end{cases}
\]

**Definition 5.** A regexp \( R \) is syntactically \( \leq \)-closed (SC) if \( R = r_1 \cdots r_k \), where \( k \geq 0 \) and each \( r_i \) is PSC. For the \( k = 0 \) case, we define \( R := \varepsilon \) by convention. If \( w \) is a string, we define an \( R \)-partition of \( w \) to be a list \( \langle w_1, \ldots, w_k \rangle \) of strings such that \( w_1 \cdots w_k = w \) and \( w_i \in L(r_i) \) for each \( 1 \leq i \leq k \). We call \( w_i \) the \( i \)th component of the \( R \)-partition.

**Observation 6.** If regexp \( R \) is SC, then \( L(R) \) is \( \leq \)-closed.

**Observation 7.** For SC \( R \) and string \( w \), \( w \in L(R) \) iff some \( R \)-partition of \( w \) exists.

**Definition 8.** Let \( r = \Sigma^* \) be an unbounded PSC regexp. We define \( \text{pref}(r) \), the **primitive refinement** of \( r \), as follows: if \( \Sigma = \{ a \} \) for some \( a \in \mathbb{N} \), then let \( \text{pref}(r) \) be the bounded regexp \( a \cup \varepsilon \); otherwise, if \( \Sigma = \{ n_1 < n_2 < \cdots < n_k \} \) for some \( k \geq 2 \), then we let
\[
\text{pref}(r) := (\Sigma - \{ n_1 \})^*(\Sigma - \{ n_2 \})^* \cdots (\Sigma - \{ n_k \})^*.
\]

To use the definition above, note that \( \text{pref}(r) \) is SC but not PSC. Also note that \( L((\text{pref}(r))^n) = L(r) \). This leads to the following definition, analogous to Definition 2:

**Definition 9.** Let \( r \) be an unbounded PSC regexp, and let \( w \in L(r) \) be a string. Define
\[
m_r(w) := \min\{ n \in \mathbb{N} \mid w \in L((\text{pref}(r))^n) \}.
\]

There is a nice connection between Definitions 2 and 9, given by the following Lemma:

**Lemma 10.** For any unbounded PSC regexp \( r = \Sigma^* \) and any string \( w \in L(r) \),
\[
m_r(w) = \begin{cases} 
\ell_\Sigma(w) & \text{if } |\Sigma| = 1, \\
\ell_\Sigma(w) + 1 & \text{if } |\Sigma| \geq 2.
\end{cases}
\]

**Proof.** First, if \( |\Sigma| = 1 \), then \( \text{pref}(r) = a \cup \varepsilon \) and \( \sigma_\Sigma = a \), where \( \Sigma = \{ a \} \). Then clearly,
\[
m_r(w) = |w| = \ell_\Sigma(w).
\]

Second, suppose that \( \Sigma = \{ n_1 < \cdots < n_k \} \) with \( k \geq 2 \), so that \( \sigma_\Sigma = n_1 \cdots n_k \) and \( \text{pref}(r) = \Sigma_1^* \cdots \Sigma_k^* \) from (1), where we set \( \Sigma_i = \Sigma - \{ n_i \} \) for \( 1 \leq i \leq k \). Let \( m = m_r(w) \), and let \( P = \langle w_1,1, \ldots, w_{1,k}, w_2,1, \ldots, w_{2,k}, \ldots, w_{m,1}, \ldots, w_{m,k} \rangle \) be any \( (\text{pref}(r))^m \)-partition of \( w \) (at least one such partition exists by Observation 7). We have that each \( w_{i,j} \in L(\Sigma_j^*) \). If \( (\sigma_\Sigma)^t \leq w \) for some \( t \geq 0 \), then there is some monotone nondecreasing map \( p : \{ 1, \ldots, \ell k \} \rightarrow \{ 1, \ldots, mk \} \) such that the \( t \)th symbol of \( (\sigma_\Sigma)^t \) occurs in the \( p(t) \)th component of \( P \). Now we must have \( p(t) \neq t \) for all \( 1 \leq t \leq \ell k \); writing \( t = qk + s \) for some \( 1 \leq s \leq k \), we have that the \( t \)th symbol of \( (\sigma_\Sigma)^t \) is \( n_s \), but the \( t \)th component of \( P \) is \( w_{q+1,s} \in L(\Sigma_s^*) \), and \( n_s \notin \Sigma_s \). Thus the \( t \)th symbol in \( (\sigma_\Sigma)^t \) does not occur in the \( t \)th
component of $P$, and so $t \neq p(t)$. Now it follows from the monotonicity of $p$ that $p(t) > t$ for all $t$. In particular, $\ell_k < p(\ell_k) \leq mk$, and so $\ell < m$. This shows that $n_r(w) \geq \ell(w) + 1$.

Let $m$ be as in the previous paragraph. We build a particular $(\text{pref}(r))^m$-partition $P_{\text{greedy}} = \langle w_1, \ldots, w_{1,k}, w_{2,1}, \ldots, w_{2,k}, \ldots, w_{m,1}, \ldots, w_{m,k} \rangle$ of $w$ by the greedy algorithm below. In the algorithm, for integers $1 \leq i \leq m$ and $1 \leq j \leq k$ we let

$$(i, j)' = \begin{cases} (i, j + 1) & \text{if } j < k, \\ (i + 1, 1) & \text{otherwise}. \end{cases}$$

This is the successor operation in the lexicographical ordering on the pairs $(i, j)$ with $1 \leq j \leq k$: $(i_1, j_1) < (i_2, j_2)$ if either $i_1 < i_2$ or $i_1 = i_2$ and $j_1 < j_2$.

$$(i, j) \leftarrow (1, 1)$$
While $i \leq m$

Let $w_{i,j}$ be the longest prefix of $w$ in $\Sigma_j$.

Remove prefix $w_{i,j}$ from $w$.

$$(i, j) \leftarrow (i, j)'$$
End while.

Since some $(\text{pref}(r))^m$-partition of $w$ exists, this algorithm will clearly also produce a $(\text{pref}(r))^m$-partition of $w$, i.e., the while-loop terminates with $w = \varepsilon$. Furthermore, $w$ does not become $\varepsilon$ until the end of the $(m,1)$-iteration of the loop at the earliest; otherwise, the algorithm would produce a $(\text{pref}(r))^{m-1}$-partition of $w$, contradicting the minimality of $m$. Finally, for all $(i, j)$ lexicographically between $(1, 1)$ and $(m-1, k)$ inclusive, letting $(i', j') = (i, j)'$, we have that $w_{i', j'}$ starts with $n_j$. This follows immediately from the greediness (maximum length) of the choice of $w_{i,j}$. Therefore, we have $\sigma_\Sigma$ is a subsequence of each of the strings $(w_{1,1} \cdots w_{1,k}), (w_{2,1} \cdots w_{2,k}), \ldots, (w_{m-1,1} \cdots w_{m-1,k})$, and so $(\sigma_\Sigma)^{m-1} \preceq w$, which proves that $n_r(w) \leq \ell(w) + 1$.

**Definition 11.** Let $R = r_1 \cdots r_k$ and $S$ be two SC regexps, where each $r_i$ is PSC. We say that $S$ is a one-step refinement of $R$ if $S$ results from either

- removing some bounded $r_i$ from $R$, or
- replacing some unbounded $r_i$ in $R$ by $(\text{pref}(r_i))^n$ for some $n \in \mathbb{N}$.

We say that $S$ is a refinement of $R$ (and write $S < R$) if $S$ results from $R$ through a sequence of one or more one-step refinements.

One may note that if $S < R$, then $L(S) \subseteq L(R)$, although it is not important to the main proof.

**Lemma 12.** The relation $<$ of Definition 11 is a well-founded partial order on the set of SC regexps (of height at most $\omega^\omega$).
Proof. Let $R = r_1 \cdots r_k$ be an SC regexp, and let $e_1 \geq e_2 \geq \cdots \geq e_k$ be the ranks of all the $r_i$, arranged in nonincreasing order, counting duplicates. Define the ordinal

$$\text{ord}(R) := \omega^{e_1} + \omega^{e_2} + \cdots + \omega^{e_k},$$

which is in Cantor normal form and always less than $\omega^{e_i}$. If $R = \varepsilon$, then $\text{ord}(R) := 0$ by convention. Let $S$ be an SC regexp. Then it is clear that $S < R$ implies $\text{ord}(S) < \text{ord}(R)$, because the ord of any one-step refinement of $R$ results from either removing some addend $\omega^0 = 1$ or replacing some addend $\omega^e$ for some positive $e$ (the rightmost with exponent $e$) in the ordinal sum of $\text{ord}(R)$ with the ordinal $\omega^{e-1} \cdot n$, for some $n < \omega$, resulting in a strictly smaller ordinal. From this the lemma follows. \qed

2 Main Proofs

The following lemma is key to proving Theorem 1.

Lemma 13 (Key Lemma). Let $R = r_1 \cdots r_k$ be a SC regexp where at least one of the $r_i$ is unbounded. Suppose $L \subseteq L(R)$ is $\preceq$-closed. Then either

1. $L = L(R)$ or

2. there exist refinements $S_1, \ldots, S_k < R$ such that $L \subseteq \bigcup_{i=1}^k L(S_i)$.

Before proving Lemma 13, we see how it is used to prove Theorem 1.

Proof of Theorem 1. Let $L \subseteq L(\Sigma^*)$ be $\preceq$-closed. We prove by induction on the refinement relation that: for any SC regexp $R$, if $L \subseteq L(R)$ then $L$ is regular. The theorem follows by setting $R = \Sigma^*$. Fix $R = r_1 \cdots r_k$, and suppose that $L \subseteq L(R)$. If all of the $r_i$ are bounded, then $L(R)$ is finite and hence $L$ is regular. Now assume that at least one $r_i$ is unbounded and that the statement holds for all $S < R$. If $L = L(R)$, then $L$ is certainly regular, since $R$ is a regexp. If $L \neq L(R)$, then by Lemma 13 there are $S_1, \ldots, S_k < R$ with $L \subseteq \bigcup_{i=1}^k L(S_i)$. Each $L \cap L(S_i)$ is $\preceq$-closed (being the intersection of two $\preceq$-closed languages) and hence regular by the inductive hypothesis. But then,

$$L = L \cap \bigcup_{i=1}^k L(S_i) = \bigcup_{i=1}^k (L \cap L(S_i)),$$

and so $L$ is regular. \qed

Proof of Lemma 13. Fix $R$ and $L$ as in the statement of the lemma. Whether Case 1 or Case 2 holds hinges on whether or not a certain quantity associated with each string in $L(R)$ is unbounded when taken over all strings in $L$.

For any string $w \in L(R)$ and any $R$-partition $P = \langle w_1, \ldots, w_k \rangle$ of $w$, define

$$M_P^{bd}(w) := \min_{i: r_i \text{ is bounded}} |w_i|,$$

for any string $w \in L(R)$ and any $R$-partition $P = \langle w_1, \ldots, w_k \rangle$ of $w$, define

$$M_P^{bd}(w) := \min_{i: r_i \text{ is bounded}} |w_i|,$$
and define
\[ M_n^\text{unbd}(w) := \min_{i: \text{r}_i \text{ is unbounded}} m_{r_i}(w_i). \] (3)

In (2), for any bounded \( r_i \), we have \( w_i \in L(r_i) \) and thus \( |w_i| \in \{0, 1\} \). If there is no bounded \( r_i \), we'll take the minimum to be 1 by default.

Now define
\[ M(w) := \max_{P: P \text{ is an } R\text{-partition of } w} M_P^\text{bd}(w) \cdot M_P^\text{unbd}(w). \] (4)

We will show that if
\[ \limsup_{w \in L} M(w) = \infty, \] (5)

then Case 1 of the lemma holds. Otherwise, Case 2 holds.

Suppose that (5) holds. Let \( x \in L(R) \) be arbitrary. Then there is a \( w \in L \) such that \( |x| \leq M(w) \). For this \( w \) there is an \( R\)-partition \( P = \langle w_1, \ldots, w_k \rangle \) of \( w \) such that \( M_P^\text{bd}(w) = 1 \) and \( M_P^\text{unbd}(w) > |x| \). Let \( \langle x_1, \ldots, x_k \rangle \) be some \( R\)-partition of \( x \). For all \( 1 \leq i \leq k \), we then have

- \( |x_i| \leq 1 = |w_i| \) if \( r_i \) is bounded, and
- \( |x_i| \leq |x| \leq m_{r_i}(w_i) - 1 \leq \ell_r(w_i) \) if \( r_i = \Gamma^* \) for some alphabet \( \Gamma \).

(The last inequality of the second item follows from Lemma 10). In either case, we have \( x_i \preceq w_i \) (the second case following from Observation 3), and thus \( x \preceq w \). Since \( w \in L \) and \( L \) is \( \preceq \)-closed, we have \( x \in L \). Since \( x \in L(R) \) was arbitrary, this proves that \( L = L(R) \), which is Case 1 of the lemma.

Now suppose that (5) does not hold. This means there is a finite bound \( B \) such that \( M(w) \leq B \) for all \( w \in L \). So for any \( w \in L \) and any \( R\)-partition \( P = \langle w_1, \ldots, w_k \rangle \) of \( w \), either \( M_P^\text{bd}(w) = 0 \) or \( M_P^\text{unbd}(w) \leq B \). Suppose \( M_P^\text{bd}(w) = 0 \). Then \( w_i = \varepsilon \) for some \( i \) where \( r_i \) is bounded. Let \( S_i \) be the one-step refinement of \( R \) obtained by removing \( r_i \) from \( R \). Then clearly, \( w \in L(S_i) \). Now suppose \( M_P^\text{unbd}(w) \leq B \), so that there is some unbounded \( r_j \) such that \( m_{r_j}(w_j) \leq B \). This means that \( w_j \in L((\text{pref}(r_j))^B) \) by Definition 9. Let \( S_j \) be the one-step refinement obtained from \( R \) by replacing \( r_j \) with \( (\text{pref}(r_j))^B \). Then clearly again, \( w \in L(S_j) \). In general, we define, for all \( 1 \leq i \leq k \),

\[ S_i = \begin{cases} r_1 \cdots r_{i-1} r_{i+1} \cdots r_k & \text{if } r_i \text{ is bounded}, \\ r_1 \cdots r_{i-1} (\text{pref}(r_i))^B r_{i+1} \cdots r_k & \text{otherwise}. \end{cases} \]

We have shown that there is always an \( i \) for which \( w \in L(S_i) \). Since \( w \in L \) was arbitrary, Case 2 of the lemma holds. \( \square \)